PATENT APPLICATION
ATTORNEY DOCKET NO. 09669/043001

APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: **USB DEVICE**

APPLICANT: **Nicolas DRABCZUK**

"EXPRESS MAIL" Mailing Label Number: **EV535680165US**

Date of Deposit: **January 13, 2004**

USB DEVICE

BACKGROUND OF INVENTION

Field of the Invention

[0001] The invention relates to a system comprising a first device arranged to communicate with a second device. The first device can be, for example, a USB host. The second device can be, for example, a USB device, which communicates with the USB host via a USB bus using the USB protocol.

Background Art

[0002] The system comprises a USB host, which is connected to various USB devices via a USB bus. The USB host communicates with the USB devices with the USB protocol. The USB protocol allows connecting several USB devices on the same USB bus using a system of time-sharing based on addressed devices. The USB protocol is organized as a master/slave architecture, the USB host is thus responsible of the time-sharing management.

[0003] The USB host may comprise various applications. One or several services may be needed to run an application. An application uses one or several drivers to access and use the associated services. The drivers may be on the USB host.

[0004] The USB device may comprise various services, in particular those, which are needed by the applications of the USB host. A service may be offered, for example, at the device level (standard USB device), or at the interface level (composite USB device).

[0005] The USB device is organized into several levels a device level, a configuration level, an interface level and an endpoint level. Each level is represented by different USB descriptors:

- A device descriptor describing the overall device. The device descriptor may be associated to one or more configuration descriptors.

- A configuration descriptor describing the electrical characteristics of the USB device, or of a part of the USB device. The configuration descriptor may be associated to one or more interface descriptors.
- An interface descriptor describing a particular service of the USB device. An interface may contain one or more alternate settings. The interface descriptor may be associated to zero or more endpoint descriptors.
- An endpoint descriptor describing a communication channel used by the service defined by the interface descriptor.

[0006] In a plugging step, the USB device is plugged onto a USB port of the USB host.

[0007] In an enumeration step, all the USB descriptors are then retrieved from the USB device to the host device. The enumeration step is triggered off with the modification of the voltage level on the line D+ or D- (depending on the USB device speed) due to a pull-up resistor present in the USB device on one of the lines.

[0008] In a loading step, the USB Host then uses the descriptors to load all the drivers of the USB device._The number of drivers loaded depends on the number of different services present in the USB device. For example, if the USB device is at the same time a scanner and a printer the USB device will have to present two interfaces during the enumeration step. In that case, two drivers, one associated to the scanner interface, and the other one associated to the printer interface, will be loaded. A main driver associated to the device itself could also be loaded.

[0009] US 2001/0027500 discloses a data transmission system comprising a host, a controller connected to the host via a plug and play compatible bus, and a plurality of functions provided by the connected controller, that exceed in number the maximum end points that the controller can support, wherein the controller selects a predetermined number of functions from among the plurality of functions and enumerates, as end points, the functions that match in number the maximum end points, and wherein thereafter, the controller replaces with a different function one of the functions that constitutes the end-points, and re-enumerates the endpoints.

SUMMARY OF INVENTION

[0010] An object of the invention is to reduce the costs

[0011] According to one aspect of the invention, a method of configuring a system comprising a main device and an auxiliary device arranged to co-operate with each other, the main device being arranged to handle one or more functionalities, the auxiliary device being arranged to effect one or more functionalities, characterised in that the method comprises an adaptation step, in which the auxiliary device is made to hide from the main device at least those of its functionalities for which the main device is not arranged to handle.

[0012] The first device can be, for example, a USB host. The second device can be, for example, a USB device. The functionalities for which the main device is not arranged to handle will be hidden from the main device. In particular, the invention allows to mass-produce auxiliary devices arranged to effect the same standard set of functionalities. Thus the invention allows a reduction of the costs.

BRIEF DESCRIPTION OF DRAWINGS

[0013] Fig. 1 illustrates a system comprising a USB host and a USB device.

[0014] Fig.2 illustrates the structure of the USB device.

[0015] Fig. 3 illustrates a method of using the system; and

[0016] Fig. 4 illustrates a system comprising a USB host and a USB device.

[0017] Fig. 5 illustrates a method of using the system.

DETAILED DESCRIPTION

[0018] As illustrated in figure 1, the invention will be explained in the context of a system using the USB protocol. The system comprises a USB host, which is connected to a USB device via a USB bus. The USB host communicates with the USB devices with the USB protocol.

[0019] The communication between the USB host and the USB device can be made according four different transfer modes:

- a control transfer, offering mainly a delivery and data integrity guarantee,
- an interrupt transfer, offering mainly a periodicity and data integrity guarantee,
- a bulk transfer, offering mainly a data integrity guarantee, and a possibly good data rate,
- a isochronous transfer, offering mainly a bandwidth guarantee.

[0020] Two of these four modes require a bandwidth reservation, which is accorded or not by the USB host after an enumeration phase, depending on the bandwidth already reserved by other USB devices, which are plugged onto the USB bus.

[0021] The USB host comprises various applications (A1, A2). One or several services (S1, S2) are needed to run an application (A1). The application A1 needs, for example, to use the service S1 and the service S2. The services (S1, S2, S3, S4) are located on the USB device. To use a specific service S1, an application A1 may use a driver D1. The USB host also comprises a standard application (A0) associated with a standard driver (D0). Advantageously the standard application (A0) is implemented on a big number of USB hosts.

[0022] The USB device comprises various services (S1, S2, S3 and S4) in particular those (S1, S2, S3), which are needed by the applications (A1, A2) of the USB host. As illustrated in figure 2, a service (S) can be offered, for example, at the device level, or at the interface level (S1,S2,S3,S4). The USB device also comprises a standard service (S0). Advantageously the standard service (S0) is implemented on a big number of USB devices.

[0023] As illustrated in figure 3, in a connecting step CON, the USB device is connected to a USB port of a USB host.

[0024] In a first checking step CHECK1, the USB device checks whether a negotiation flag is activated or not.

If not:

- in a first enumerating step ENUM1, the USB host will enumerate the USB device. In other words, as illustrated in figure 2, the USB host will retrieve from the USB device to the USB host only the descriptors (I) associated to the standard service S0,
- in a loading step LOAD, the standard driver D0 is loaded into an active memory of the host,
- in a negotiation step NEGO, the standard application A0 negotiates the services (S1, S2, S3) to activate. The negotiating step comprises the following sub-steps:
  - a receiving step, in which the standard application A0 receives from the standard service S0 a first list of all the different services (S1, S2, S3, S4) which are available on the USB device,
  - a comparing step, in which the standard application compares the first list of all the different services (S1, S2, S3, S4) which are available on the USB device with a second list of the services (S1, S2, S3) needed by the applications (A1,A2) of the USB host to deduce the services to be activated (S1, S2, S3) on the USB device,
  - a service activating step, in which the USB device activates the services to be activated, for example, by disconnecting and reconnecting the USB device to the USB host.
- In an flag activating step ACTIV, the negotiation flag is activated.
- In an initialization step INIT, the USB device removes its pull-up resistor in order to detach itself and then re-attach itself.

[0025] In a second checking step CHECK2, the USB device checks whether the negotiation flag is activated or not.

If yes:

- in a deactivating step DEACTIV, the negotiation flag is deactivated,
- in a second enumerating step ENUM2, the USB host enumerates the USB device. As illustrated in figure 2, only the descriptors (II) associated to the

services (S1, S2, S3) which have been activated and the descriptor associated to the standard service (S0) will be retrieved,

- in a second loading step LOAD2, the standard driver D0 and the drivers associated to the services (S1, S2, S3) which have been activated are loaded into the active memory of the USB host.

[0026] The USB device is now ready for use.

[0027] According to an advantage of the invention, if a new service has to be added on the USB device, the standard service (S0) does not change and therefore the standard application (A0) does not change. The invention thus allows a reduction of the cost.

[0028] As illustrated in figure 4 and 5, if the USB device is already plugged, and the user starts a new application (A3), which requires a new service (S4), which is not activated in the USB device, the standard application (A0) can negotiate the activation of the new service (S4) in a new negotiating step.

[0029] In an opening step OPEN, user opens a new application (A3) requiring a service (S4), which is not available in the current configuration of the USB device.

[0030] In a negotiating step NEGO, the USB host activates the service (S4).

[0031] In a flag activating step ACTIV, the USB device activates the negotiation flag.

[0032] In an initialization step INIT, the USB device removes its pull-up resistor in order to detach itself and then re-attach

[0033] In a checking step CHECK, the USB device checks whether the negotiation flag is activated or not.

If yes:

- in a deactivating step DEACTIV, the negotiation flag is deactivated,
- in an enumerating step ENUM, the USB host enumerates the USB device. As illustrated in figure 2, only the descriptors (III) associated to the services (S1, S2, S3, S4) which have been activated and the descriptor associated to the standard service (S0) will be retrieved,
- in a loading step LOAD, the standard driver D0 and the drivers associated

to the services (S1, S2, S3,S4) which have been activated are loaded into the active memory of the USB host.

[0034] The device is ready for use.

[0035] The description hereinbefore illustrates the following features:

[0036] The invention concerns a method of configuring a system. The system comprises a main device and an auxiliary device. The main device and the auxiliary device are arranged to co-operate with each other. The main device is arranged to handle one or more functionalities. The auxiliary device is arranged to effect one or more functionalities. The method is characterised in that the method comprises an adaptation step, in which the auxiliary device is made to hide from the main device at least those of its functionalities that the main device cannot handle.

[0037] The first device can be, for example, a USB host in particular a computer, a PDA or GSM.

[0038] The second device can be, for example, a USB device in particular a smart card or more generally any device that can be personalized, for example, a PDA or GSM.

[0039] The USB device can be, for example, a Smart Card comprising three different services:

- Keys and rights management (APDU command transport) as service [S0],
- Document signature as service [S1],
- Data streaming application (DRM) as service [S2].

[0040] The Smart Card can be used in different USB hosts non-exhaustively listed hereafter:

- Corporate Personal Computers running Windows XP as environment [E1],
- Home Personal Computers running Windows XP as environment [E2],
- GSM (also as USB On-The-Go device) as environment [E3],
- PDA (as USB On-The-Go device) as environment [E4].

[0041] For all these USB hosts, the services that can be accessed could be:

[S0] and [S1] for [E1], because the user is not administrator of the

machine, and can not install a new driver,

[S0], [S1] and [S2] for [E2], because the user is administrator of the machine and can install any service available,

[S0] only for [E3], for memory or consumption economy reasons,

[S0] only for [E4], because the host cannot be personalized and only the driver for [S1] is available.

[0042] The services located on the USB device are not necessarily related to the USB device itself. The services can be vendor specific or not. For example; a smartcard (USB device) may have a modem service, so that the smartcard can be seen as a distant terminal through a modem connection.

[0043] It should be clear that the invention is not limited to devices communicating using the USB protocol. Other protocol like, for example, firewire based protocol may be used.

[0044] It should be clear that the invention is not limited to devices communicating according to a master/salve protocol.

[0045] In the comparing step, the first list of services and the second list of services have been used to deduce the services to be activated on the USB device. More generally any other set of data defining the services can be used to deduce the services to be activated. It can be, for example, a set of data identifying various services and giving the bandwidth needed by these services.

[0046] In the USB context, new USB Device Classes can be defined. All the USB hosts must contain default drivers for the defined classes. When a USB Device Class is modified because, for example, some new functionality is added, it is difficult to update all the USB hosts. The above-described invention allows adding some functionality to a device without having to modify the Device Class, and so, without modifying the standard drivers of the USB host.